

Décembre 2015
Communications numériques
TS226



TS 226 : ADS-B

Thibaut Gourdel
Perrot Rémi
tgourdel@enseirb-matmeca.fr
rperrot@enseirb-matmeca.fr

Contents

I	Couche physique ADS-B	2
1	Travail préliminaire	2
2	Synchronisation en temps et en fréquence	11
II	Traitement / décodage de signaux réels	12
1	Structure des trames ADS-B	13
2	Travail à réaliser	13
III	Temps réel	13
1	Optimisation temps réel	14
1.1	Détermination du CRC	14
1.2	Registre	14
1.3	Convolutions	15
2	Correction d'une erreur	15
3	Vitesse et direction des avions	15
4	Figures de résultat	16

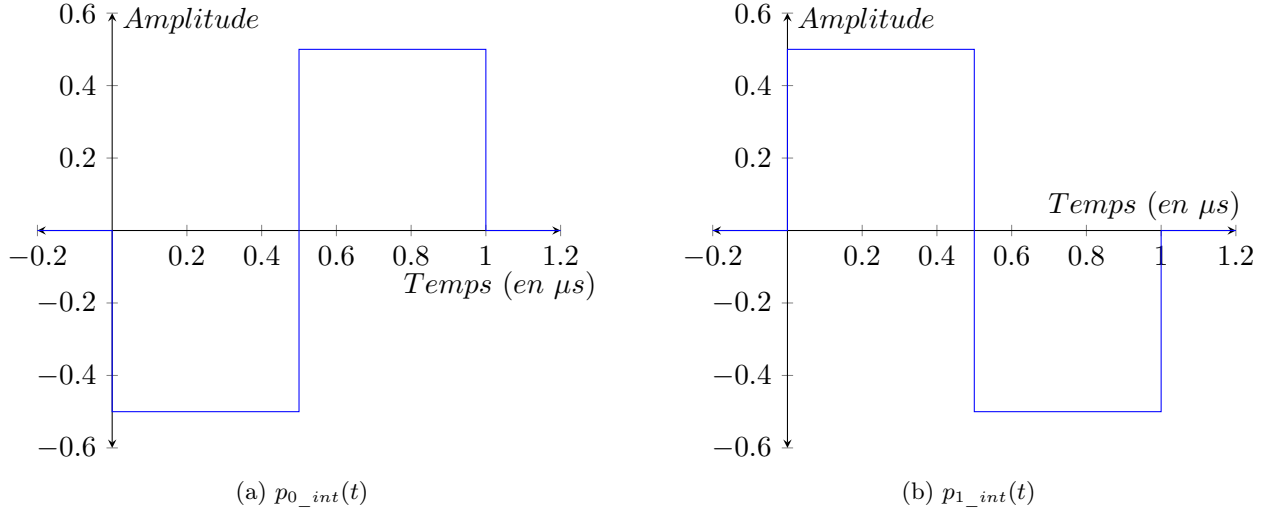


Figure 1: Représentation de $p_{b_k_int}(t)$

Part I

Couche physique ADS-B

1 Travail préliminaire

Question 1 On a le signal $s_l(t) = \sum_{k \in \mathbb{Z}} p_{b_k}(t - kT_s)$, avec $p_{b_k}(t)$ défini par $p_0(t)$ pour $b_k = 0$, et $p_1(t)$ pour $b_k = 1$.

Soit les fonctions intermédiaires (définies uniquement entre 0 et T_s - nulles sinon) :

$$p_{0_int}(t) = p_0(t) - 0.5 \quad (1)$$

$$p_{1_int}(t) = p_1(t) - 0.5 \quad (2)$$

On obtient les fonctions de la figure 1.

Ainsi, on peut définir la fonction $s_{int}(t)$, elle aussi intermédiaire :

$$\begin{aligned} s_{int}(t) &= s_l(t) - 0,5 = \left(\sum_{k \in \mathbb{Z}} p_{b_k}(t - kT_s) \right) - 0,5 \\ &= \sum_{k \in \mathbb{Z}} p_{b_k_int}(t - kT_s) \end{aligned} \quad (3)$$

On remarque aux sous-figures de la figure 1 que :

$$\begin{cases} \text{Pour } b_k=0 & \rightarrow p_{0_int}(t) = p(t) \\ \text{Pour } b_k=1 & \rightarrow p_{1_int}(t) = -p(t) \end{cases} \quad (4)$$

Grâce aux symboles A_k ,

$$A_k = \begin{cases} 1 & \text{pour } b_k = 0 \\ -1 & \text{pour } b_k = 1 \end{cases} \quad (5)$$

On peut directement remplacer dans l'équation 3 $p_{b_k_int}(t)$ par $p(t)$, en utilisant A_k :

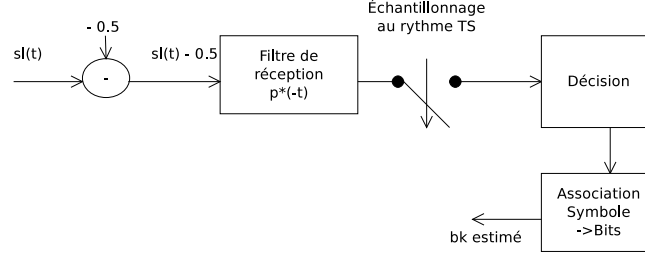


Figure 2: Architecture pour une modulation avec $s_l(t) - 0,5$

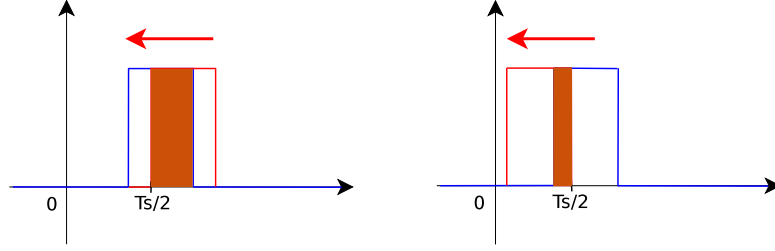


Figure 3: Schéma de la convolution des couples de filtres

$$s_{int}(t) = \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) \quad (6)$$

Soit finalement $s_l(t) = s_{int}(t) + 0,5 = 0,5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)$.

Modulation obtenue : La modulation obtenue avec $s_l(t) - 0,5$ nous mets face à un code biphasé (Manchester), standardisé par l'IEEE 802.3¹. Ce code est lié à une modulation en bande de base : PSK, Phase Shift Keying, où le signal est un signal carré (plutôt que - habituellement - une sinusoïde).

Avec ce code, la modulation est effectuée en faisant une convolution entre les symboles A_k et le filtre $p(t)$.

Pour retrouver les bits d'entrées, on peut donc convoluer avec le filtre adapté $p^*(-t)$. Cela a pour effet de créer une sorte de signal triangulaire, dont les points aux moments d'échantillonnage correspondent aux symboles. Voir figure 2.

Remarque Il faut garder à l'esprit que cette modulation fonctionne pour $s_l(t) - 0,5$, pour l'implémenter dans Matlab il faudra soustraire 0,5 au signal $s_l(t)$.

Question 2 $p_0^*(-t)$ et $p_1^*(-t)$ sont, respectivement, les filtres de mise en forme de $p_0(t)$ et $p_1(t)$. Ce sont les filtres adaptés ; l'utilisation de tels filtres permet de maximiser le SNR (Signal-to-Noise Ratio).

De ce fait, on minimise la probabilité qu'il y ait une erreur binaire.

Question 3 Montrons que les couples de filtres $(p_0(t), p_0^*(-t))$ et $(p_1(t), p_1^*(-t))$ vérifient le critère de Nyquist.

Le critère de Nyquist est vérifié si $\sum_{k \in \mathbb{Z}} V(f - \frac{k}{T_s}) = T_s v(0)$ c'est-à-dire si la convolution du couple est centrée en zéro et de largeur T_s .

On peut le vérifier graphiquement, voir pour cela la figure 3.

On en déduit la représentation de la convolution, donnée en figure 4.

¹<http://docwiki.cisco.com>

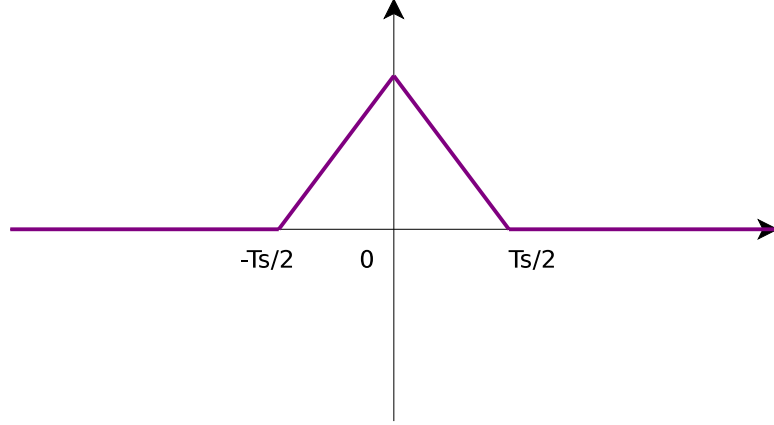


Figure 4: Représentation de la convolution d'un couple de filtre

Question 4 Calculons le moment d'ordre 1 du signal $s_l(t)$.

On peut sortir, dans 7, $p(t - kT_s)$, puisque la fonction est déterministe.

$$\begin{aligned} m_{s_l}(t) &= E[s_l(t)] = E\left[0,5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)\right] \\ &= 0,5 + \sum_{k \in \mathbb{Z}} E[A_k] p(t - kT_s) \end{aligned} \quad (7)$$

$$= 0,5 = m_{s_l} \quad (8)$$

$m_{s_l}(t)$ ne dépend donc pas de t , et de plus m_{s_l} est égal à 0,5 (puisque $E[A_k] = 0$).

Question 5 Calcul de la fonction d'auto-corrélation.

À la ligne 9, on peut enlever le terme de conjugué (*), puisque le signal est réel.

$$R_{s_l}(t, \tau) = E[s_l(t)s_l^*(t + \tau)] \quad (9)$$

$$\begin{aligned} &= E\left[\left(0,5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)\right) \left(0,5 + \sum_{k' \in \mathbb{Z}} A_{k'} p(t + \tau - k'T_s)\right)\right] \\ &= E\left[0,25 + \sum_k \sum_{k'} A_k A_{k'} p(t - kT_s) p(t + \tau - k'T_s)\right. \\ &\quad \left.+ 0,5 \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) + 0,5 \sum_{k' \in \mathbb{Z}} A_{k'} p(t + \tau - k'T_s)\right] \\ &\quad \underbrace{\hspace{10em}}_{=0, \text{ puisque } E[A_k]=0} \\ &= 0,25 + \sum_k \sum_{k'} E[A_k A_{k'}] p(t - kT_s) p(t + \tau - k'T_s) \end{aligned} \quad (10)$$

Lorsque $k \neq k'$, on a $E[A_k A_{k'}] = E[A_k] E[A_{k'}] = 0$, et donc $R_{s_l}(t, \tau) = 0,25$. Par contre, pour $k = k'$, on est face à $E[A_k^2]$, la variance des A_k (σ^2). Soit :

$$\boxed{R_{s_l}(t, \tau) = 0,25 + \sum_k \sigma^2 p(t - kT_s) p(t + \tau - kT_s)} \quad (11)$$

À noter que la variance σ^2 est égale à 1.

Question 6 On utilise les résultats des questions précédentes pour prouver la cyclo-stationnarité.

Un processus aléatoire $x(t)$ est cyclo-stationnaire, avec T_s la période de cyclo-stationnarité, lorsque $R_{xx}(t + T_s, \tau) = R_{xx}(t, \tau)$. Soit, dans notre cas :

$$R_{s_l}(t + T_s, \tau) = E \left[\left(0, 5 + \sum_{k \in \mathbb{Z}} A_k p(t + T_s - kT_s) \right) \left(0, 5 + \sum_{k' \in \mathbb{Z}} A_{k'} p(t + T_s + \tau - k'T_s) \right) \right] \quad (12)$$

Il suffit alors, dans l'équation 12, de faire le changement de variable suivant : $k = k - 1$ et $k' = k' - 1$. On trouve ainsi la relation :

$$\boxed{R_{s_l}(t + T_s, \tau) = R_{s_l}(t, \tau)} \quad (13)$$

Le processus est donc cyclo-stationnaire de période T_s (temps symbole).

Question 7 La formule de l'auto-corrélation moyennée d'un signal $s_l(t)$ est donnée ci-après :

$$\tilde{R}_{s_l}(\tau) = \frac{1}{T_s} \int_0^{T_s} R_{s_l}(t, \tau) dt \quad (14)$$

Soit, en développant (on rappelle que le symbole du conjugué disparaît en raison de la nature réelle du signal) :

$$\begin{aligned} \tilde{R}_{s_l}(\tau) &= \frac{1}{T_s} \int_0^{T_s} \left(0, 25 + \sum_k p(t - kT_s) p(t + \tau - kT_s) \right) dt \\ &= \frac{1}{T_s} \left(0, 25T_s + \int_0^{T_s} \sum_k p(t - kT_s) p(t + \tau - kT_s) dt \right) \\ &= 0, 25 + \frac{1}{T_s} \sum_k \int_0^{T_s} p(t - kT_s) p(t + \tau - kT_s) dt \end{aligned} \quad (15)$$

On pose $\nu = t - kT_s$. Ainsi, en remplaçant à la ligne 15 :

$$\begin{aligned} \tilde{R}_{s_l}(\tau) &= 0, 25 + \frac{1}{T_s} \sum_k \int_{-kT_s}^{(-k+1)T_s} p(\nu) p(\nu + \tau) d\nu \\ &= 0, 25 + \frac{1}{T_s} \int_R p(\nu) p(\nu + \tau) d\nu \end{aligned} \quad (16)$$

En observant la ligne 16, on remarque qu'il y a la présence de la fonction d'auto-corrélation de p : $R_{pp}(\tau) = \int_R p(\nu) p^*(\nu + \tau) d\nu$. Ainsi :

$$\boxed{\tilde{R}_{s_l}(\tau) = 0, 25 + \frac{1}{T_s} R_{pp}(\tau)} \quad (17)$$

Tout d'abord, calculons la fonction d'auto-corrélation de $p(\frac{t}{T_s})$. Graphiquement, on trouve l'auto-corrélation de la fonction (voir figure 5.a). On prend t/T_s pour pouvoir gérer tous les T_s , même si dans notre exemple $T_s = 1$. En la moyennant par T_s (le temps symbole donné dans l'énoncé correspond à $1\mu s$), puis en ajoutant 0,25, on trouve finalement la représentation de $\tilde{R}_{s_l}(\tau)$, présentée figure 5.b.

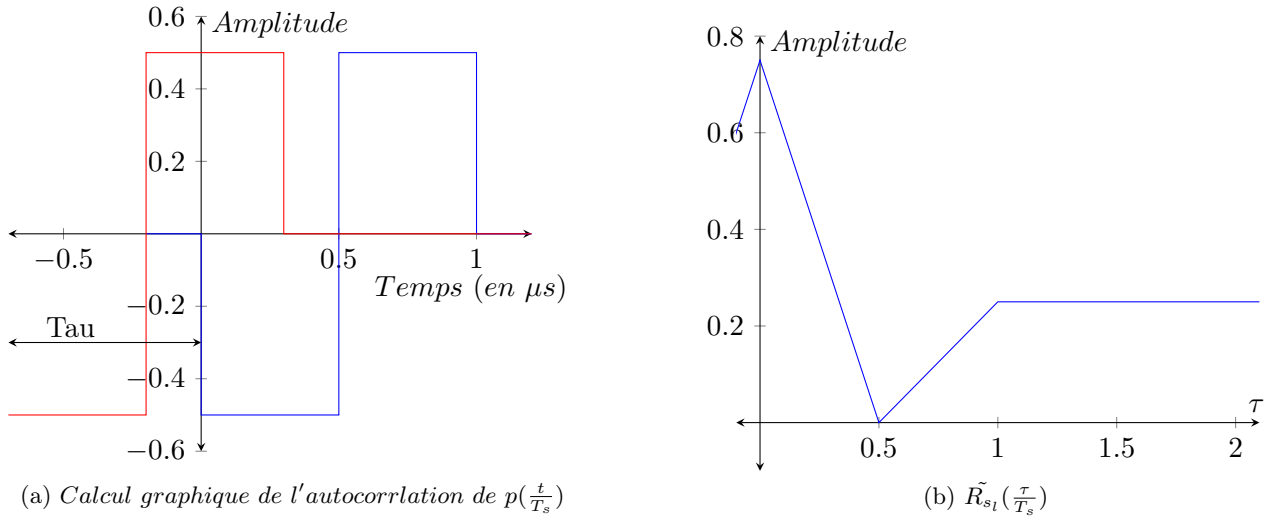


Figure 5: Calcul graphique de l'auto-corrélation moyennée

Question 8 Le signal $s_l(t)$ étant cyclo-stationnaire, on peut calculer sa densité spectrale via la transformée de Fourier de la fonction d'auto-corrélation moyennée $\tilde{R}_{s_l}(\tau)$:

$$\begin{aligned}
 \Gamma_{s_l}(f) &= \int_R \tilde{R}_{s_l}(\tau) e^{-j2\pi f\tau} d\tau \\
 &= \int_R [0.25 + \frac{1}{T_s} R_p(\tau) e^{-j2\pi f\tau}] d\tau \\
 &= \int_R 0.25 e^{-j2\pi f\tau} d\tau + \frac{1}{T_s} \int_R R_p(\tau) e^{-j2\pi f\tau} d\tau \\
 &= 0.25\delta(f) + \frac{1}{T_s} |P(f)|^2
 \end{aligned} \tag{18}$$

On remplace la transformée de Fourier de $R_p(\tau)$ par le module au carré de la transformée de Fourier de $p(t)$ dans l'équation 18, en utilisant le théorème de Wiener Khintchine.

Il nous reste donc à calculer $P(f)$.

$p(t)$ est la combinaison de deux portes (on prend bien garde à ne pas oublier T_s) : $p(t) = -0.5 \times \Pi_{0.5T_s}(t - \frac{T_s}{4}) + 0.5 \times \Pi_{0.5T_s}(t - \frac{3T_s}{4})$. On rappelle la transformée de Fourier d'une porte :

$$\mathcal{F}(\Pi_T(t)) = T \text{sinc}(\nu T) \tag{19}$$

On définit le sinus cardinal comme : $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$.

Ainsi :

$$\begin{aligned}
|P(f)|^2 &= |\mathcal{F}(-0.5 \times \Pi_{0.5T_s}(t - T_s/4) + 0.5 \times \Pi_{0.5T_s}(t - 3T_s/4))|^2 \\
&= \left| 0.5T_s \times \left(-0.5\text{sinc}(0.5T_s \times f) \times \exp(-i2\pi f \times \frac{T_s}{4}) \right. \right. \\
&\quad \left. \left. + 0.5\text{sinc}(0.5T_s \times f) \times \exp(-i2\pi f \times \frac{3T_s}{4}) \right) \right|^2 \\
&= \left| 0.25T_s \text{sinc}(0.5T_s \times f) \left(\exp(-i\pi f \times \frac{3T_s}{2}) - \exp(-i2\pi f \times \frac{T_s}{2}) \right) \right|^2 \\
&= \left| \frac{T_s \text{sinc}(0.5T_s \times f)}{4} \left(e^{-i\pi f T_s} \left(e^{-i\pi f \times 0.5T_s} - e^{+i\pi f \times 0.5T_s} \right) \right) \right|^2 \tag{20}
\end{aligned}$$

$$= \left| \frac{T_s \text{sinc}(0.5T_s \times f)}{4} \left(e^{-i\pi f T_s} \times \sin\left(\frac{1}{2}\pi f T_s\right) \times 2i \right) \right|^2 \tag{21}$$

Pour passer de 20 à 21, on utilise la formule d'Euler : $\frac{e^{ix} - e^{-ix}}{2i} = \sin(x)$. On remarque qu'à la ligne 21, on peut remplacer le module de l'exponentielle par 1. De même, $|i| = 1$. Soit finalement :

$$\begin{aligned}
|P(f)|^2 &= \left(\frac{T_s}{2} \text{sinc}\left(\frac{T_s f}{2}\right) \times \sin\left(\frac{T_s f \pi}{2}\right) \right)^2 \\
&= \left(\frac{T_s^2 f \pi}{4} \times \text{sinc}^2\left(\frac{T_s f}{2}\right) \right)^2 \\
&= \frac{T_s^4 (f \pi)^2}{16} \times \text{sinc}^4\left(\frac{T_s f}{2}\right) \tag{22}
\end{aligned}$$

D'où, ainsi, l'expression de la densité spectrale de puissance de $s_l(t)$, en 23 :

$$\begin{aligned}
\Gamma_{s_l}(f) &= 0.25\delta(f) + \frac{1}{T_s} \left(\frac{T_s^4 (f \pi)^2}{16} \times \text{sinc}^4\left(\frac{T_s f}{2}\right) \right) \\
\boxed{\Gamma_{s_l}(f) &= 0.25\delta(f) + \frac{T_s^3 (f \pi)^2}{16} \times \text{sinc}^4\left(\frac{T_s f}{2}\right)} \tag{23}
\end{aligned}$$

Question 9 On rappelle les propriétés du bruit $n_l(t)$ en bande de base :

- Il suit la loi normale $\mathcal{N}(0, \sigma_{n_l}^2)$
- Sa densité bilatérale de puissance est $\Gamma_{n_l}(f) = \frac{N_0}{2}$

De plus, rappelons la formule du filtrage (avec $G_a(f)$ la transformée de Fourier du filtre adapté $g_a(t)$) :

$$\Gamma_{n'_l}(f) = |G_a(f)|^2 \Gamma_{n_l}(f) \tag{24}$$

Enfin, d'après le théorème de Wiener Khintchine (également utilisé à la ligne 18),

$$\Gamma_{n_l}(f) = \mathcal{F}(R_{n_l}(\tau)) \tag{25}$$

On note, enfin, que, la densité spectrale du filtre adapté $p_0^*(-t)$ est $\Gamma_{p_1} = |P_1(f)|^2$, et - de même - la densité spectrale de $p_1^*(-t)$ est $\Gamma_{p_0} = |P_0(f)|^2$.

Connaissant ces informations, on peut calculer la fonction d'auto-corrélation des composantes de ce bruit après filtrage adapté par les deux filtres. Détaillons les étapes pour la première composante. On commence par calculer la densité spectrale de puissance :

$$\Gamma_{n'_{l,0}}(f) = |P_1(f)|^2 \Gamma_{n_l}(f) \quad (26)$$

Puisque $p_1(t) = \Pi_{T_s/2}(t - \frac{T_s}{4})$, on a

$$P_1(f) = \frac{T_s}{2} \text{sinc}(\pi f \frac{T_s}{2}) \exp(-j\pi f \frac{T_s}{2}) \quad (27)$$

Soit, au final,

$$\Gamma_{p_1} = \left| \frac{T_s}{2} \text{sinc}(\pi f \frac{T_s}{2}) \exp(-j\pi f \frac{T_s}{2}) \right|^2 = \frac{T_s^2}{4} \text{sinc}^2(\pi f \frac{T_s}{2}) \quad (28)$$

Dans l'énoncé, on a l'information $\Gamma_{n_l}(f) = \frac{N_0}{2}$. Soit :

$$\Gamma_{n'_{l,0}}(f) = \frac{1}{8} \times N_0 \times T_s^2 \text{sinc}^2(\pi f \frac{T_s}{2}) \quad (29)$$

D'après 25, on a :

$$\begin{aligned} R_{n'_{l,0}}(\tau) &= \mathcal{F}^{-1} \left(\Gamma_{n'_{l,0}}(f) \right) \\ &= \frac{N_0}{2} \times \mathcal{F}^{-1} \left(\left(\frac{T_s}{2} \text{sinc}(\pi f \frac{T_s}{2}) \right) \times \left(\frac{T_s}{2} \text{sinc}(\pi f \frac{T_s}{2}) \right) \right) \end{aligned}$$

$$\boxed{R_{n'_{l,0}}(\tau) = \frac{N_0}{2} \times (\Pi_{T_s/2}(t) * \Pi_{T_s/2}(t))} \quad (30)$$

De la même façon, on trouve :

$$\boxed{R_{n'_{l,1}}(\tau) = \frac{N_0}{2} \times (\Pi_{T_s/2}(t) * \Pi_{T_s/2}(t))} \quad (31)$$

Les processus aléatoires $n_{l,0'}$ et $n_{l,1'}$ proviennent tous deux d'un bruit blanc gaussien (théoriquement) centré. Après filtrage, ils restent donc gaussien et de moyennes nulles ; ils restent blancs. Seule la variance est différente si l'énergie des filtres n'est pas unitaire.

Question 10 On rappelle la formule général de P_b : $P_b = 2 \frac{M-1}{M \log_2(M)} Q \left(\frac{v(0)}{\sigma} \right)$.

Il est important de noter qu'il faut, soit adapter les résultats du cours à notre cas particulier, soit adapter notre chaîne de communication, à la réception notamment. Par exemple, dans notre cas on aurait, tel quel, y_l à la sortie du canal :

$$y_l(t) = \sum_k A_k p(t - kT_s) + 0,5 + n_l(t) \quad (32)$$

Et donc à l'échantillonnage, il faudrait prendre en compte le terme 0,5 :

$$r_l[n] = p(0)A_n + 0,5 * p^*(-nT_s) + n_l(t) * p^*(-nT_s) \quad (33)$$

On choisit d'enlever le terme 0,5 à y_l , ce qui nous permet de nous rapprocher des résultats obtenus en cours l'année dernière. Soit finalement :

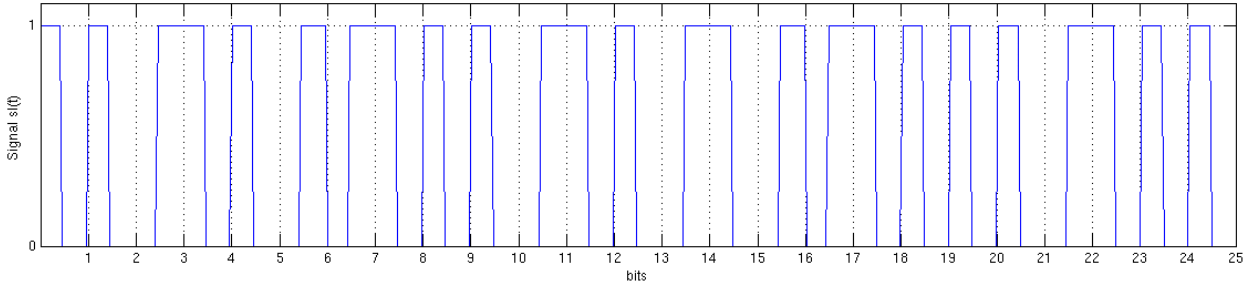


Figure 6: $s_l(t)$ sur les 25 premiers bits

$$P_b = Q\left(\sqrt{\frac{2E_g}{N_0}}\right) = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_g}{N_0}}\right) \quad (34)$$

Il semble cependant intéressant d'observer cette probabilité d'erreur en fonction de l'énergie moyenne par bit, E_b :

$$E_b = P_{moy}T_b = \frac{P_{moy}}{D_b} \quad (35)$$

Dans 35, on a P_{moy} la puissance moyenne du signal utile reçu, et T_b le temps bit. La puissance moyenne est l'intégral sur \mathcal{R} de la densité spectrale de puissance.

Or, on sait - d'après le cours - que la puissance moyenne correspond à l'expression : $P_{moy} = \frac{\sigma_A^2}{T_s}E_g$. Soit finalement :

$$\begin{aligned} E_b &= \frac{\sigma_A^2}{T_s}E_gT_b = \frac{\sigma_A^2E_g}{\log_2(M)} \\ E_g &= \frac{E_b}{\sigma_A^2} \end{aligned} \quad (36)$$

Pour 36, on a $\log_2(M) = \log_2(2) = 1$. De plus, $\sigma_A^2 = 1$. En remplaçant, on obtient donc :

$$P_b = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \quad (37)$$

Question 11 $s_l(t)$ sur les 25 premier bits est représenté figure 6.

À noter que, pour sélectionner les 25 premiers bits, il faut prendre en compte le sur-échantillonnage. On prend donc les $25 \times \frac{T_s}{T_e}$ premiers éléments du vecteur.

Question 12 Le diagramme de l'œil de $s_l(t)$ est présenté à la figure 7. L'œil est ouvert, ce qui signifie que notre signal ne souffre pas de distorsion.

Question 13 On trace la densité spectrale de puissance théorique de $s_l(t)$ sans fenêtre ni chevauchement à la figure 8.

Remarque Pendant presque toute la période du projet (sauf les deux derniers jours), notre courbe de la densité spectrale théorique était fausse, malgré tous nos efforts de résolution. Finalement ce problème a été résolu en même temps que la correction de notre code pour le temps réel. Il

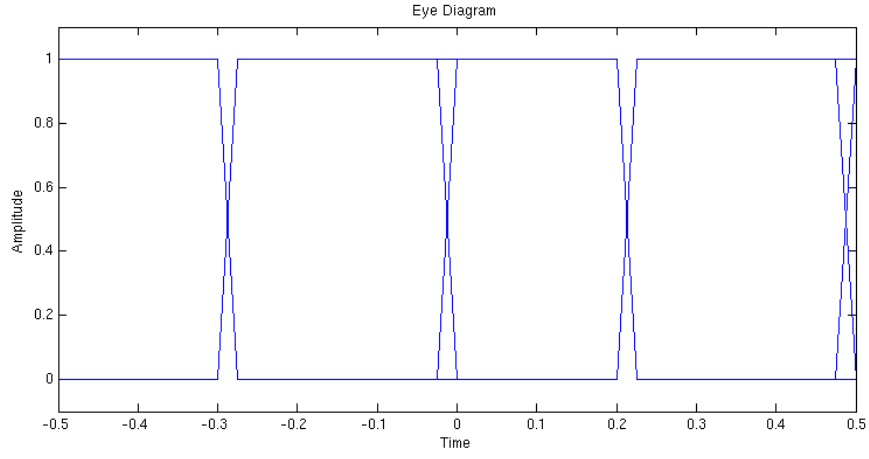


Figure 7: Diagramme de l'œil de $s_l(t)$, d'une durée de $2T_s$, des 100 premiers bits

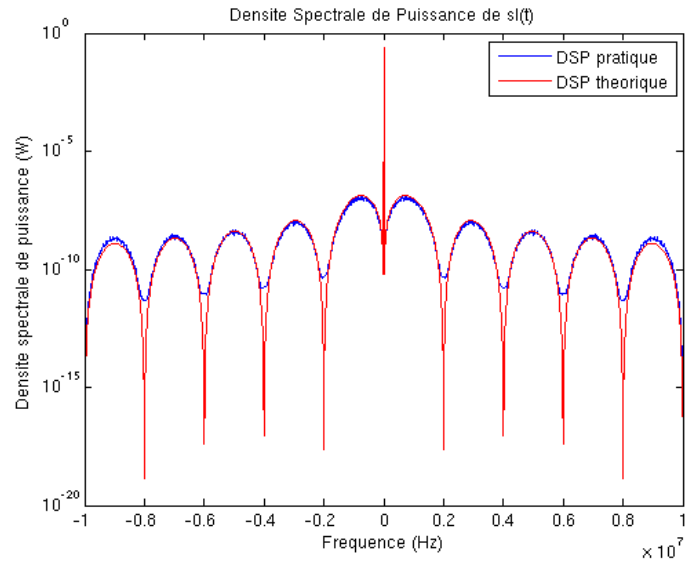


Figure 8: Densité spectrale de puissance de $s_l(t)$

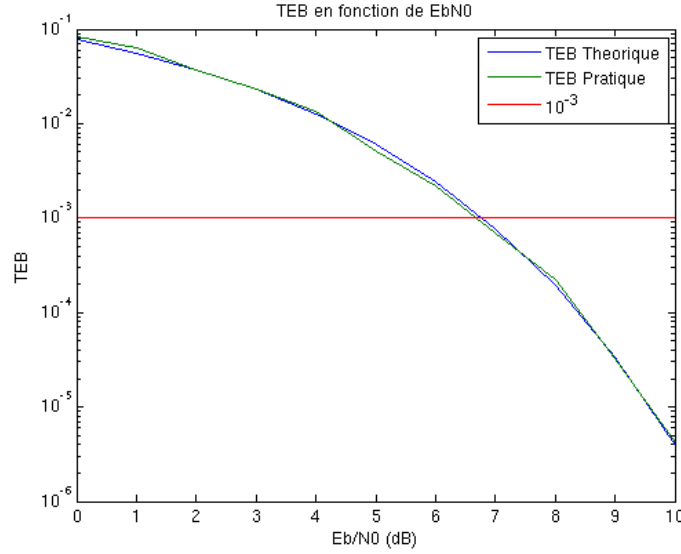


Figure 9: Taux d'Erreur Binaire en fonction du rapport signal sur bruit

s'est avéré qu'une erreur s'était glissée dans la définition de notre filtre. En lieu et place de la ligne :

`p = [-ones(1, Fse/2) ones(1, Fse/2)] / 2;`

Nous avons (un équivalent de) la commande suivante :

`p = [-ones(1, Fse/2 - 1) ones(1, Fse/2 + 1)] / 2;`

Ce qui ne permettait pas à notre DSP de se calquer sur la version théorique.

Question 14 Pour calculer le taux d'erreur binaire, on se base sur nos fichiers de l'année dernière (TS113). On trace également la courbe d'erreur théorique (équation 37, page 9), visible à la figure 9. Plutôt qu'une boucle avec x itération, on itère - pour un rapport signal sur bruit donné - jusqu'à atteindre un nombre d'erreurs rencontrées supérieur à 100. La courbe théorique suit la courbe pratique, ce qui suggère une chaîne de communication correcte.

2 Synchronisation en temps et en fréquence

Question 15 On a :

$$\rho(\delta'_t, \delta'_f) = \frac{\int_{\delta'_t}^{\delta'_t + T_p} y_l(t) s_p^*(t - \delta'_t) e^{-j2\pi\delta'_f t} dt}{\sqrt{\int_0^{T_p} |s_p(t)|^2 dt} \cdot \sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |y_l(t)|^2 dt}} \quad (38)$$

Nous allons montrer que $|\rho(\delta'_t, \delta'_f)| \leq 1$.

L'inégalité de Cauchy-Schwarz est donnée en équation 39 :

$$\left| \int_R u(t) v^*(t) dt \right| \leq \sqrt{\int_R |u(t)|^2 dt} \sqrt{\int_R |v(t)|^2 dt} \quad (39)$$

Tout d'abord, posons $u(t) = y_l(t)$ et $v^*(t) = s_p^*(t - \delta'_t) e^{-j2\pi\delta'_f t}$. Ainsi :

$$\left| \int_{\delta'_t}^{\delta'_t+T_p} y_l(t) s_p^*(t - \delta'_t) e^{-j2\pi\delta'_f t} dt \right| \leq \sqrt{\int_{\delta'_t}^{\delta'_t+T_p} |s_p(t - \delta'_t)|^2 dt} \cdot \sqrt{\int_{\delta'_t}^{\delta'_t+T_p} |y_l(t)|^2 dt} \quad (40)$$

Effectuons un changement de variable pour la première racine carré du second membre : $t' = t - \delta'_t$. D'où

$$\left| \int_{\delta'_t}^{\delta'_t+T_p} y_l(t) s_p^*(t - \delta'_t) e^{-j2\pi\delta'_f t} dt \right| \leq \sqrt{\int_0^{T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta'_t}^{\delta'_t+T_p} |y_l(t)|^2 dt} \quad (41)$$

On a donc bien le résultat de l'énoncé.

$$\boxed{|\rho(\delta'_t, \delta'_f)| \leq 1} \quad (42)$$

Question 16 Pour déterminer l'ordre de grandeur du décalage de fréquence Doppler d'un avion se déplaçant à 900km/h, on utilise la formule suivante :

$$f_{recepteur} = \frac{f_{source}}{1 - \frac{V_{source}}{V_{phase}}} \quad (43)$$

où f_{source} et $f_{recepteur}$ sont respectivement les fréquences des signaux émetteur et récepteur, V_{source} et V_{phase} sont respectivement la vitesse de la source et la vitesse de phase du signal émis. Faisons l'application numérique dans le cas de l'exemple :

$$\Delta f = f_{recepteur} - f_{source} = 1,09.10^9 - \frac{1,09.10^9}{1 - \frac{250}{3.10^8}} \quad (44)$$

$$\boxed{\Delta f \approx 900Hz} \quad (45)$$

Pour un avion se déplaçant à 900km/h, l'ordre de grandeur du décalage introduit par l'effet Doppler est d'environ 900 Hz.

C'est pour cela que dans la suite du projet il nous est demandé de tester la corrélation pour toutes les fréquences de -1kHz à 1kHz.

Question 17 En observant l'équation 38 (page 11), on remarque qu'on est face à une convolution entre le préambule inversé temporellement (utilisation de `fliplr`) d'un côté, et le signal reçu multiplié par l'exponentielle de l'autre. L'utilisation de la fonction `conv` plutôt que `xcorr` permet de gagner de précieuses secondes pour le temps réel. De même, plutôt qu'une boucle, on met en place une convolution pour le dénominateur. Pour un buffer de 0,5s, on passe d'une dizaine de secondes d'exécution à 0,3s.

Cette amélioration est mise en place dans la fonction pour le temps réel `indices_fin_preamb.m` (qui renvoie les indices de début de chaque trame potentielle).

Notre taux d'erreur binaire pratique, présenté en figure 10, ne suit pas exactement la courbe théorique. Ceci est dû au nombre d'itérations effectuées pour le calcul : comme pour la Question 14 (page 11), on itère jusqu'à obtenir 100 erreurs. Cependant, avec les décalages fréquentiels et temporels, cette valeur ne semble pas assez élevée pour se calquer exactement sur la courbe de théorique.

Part II

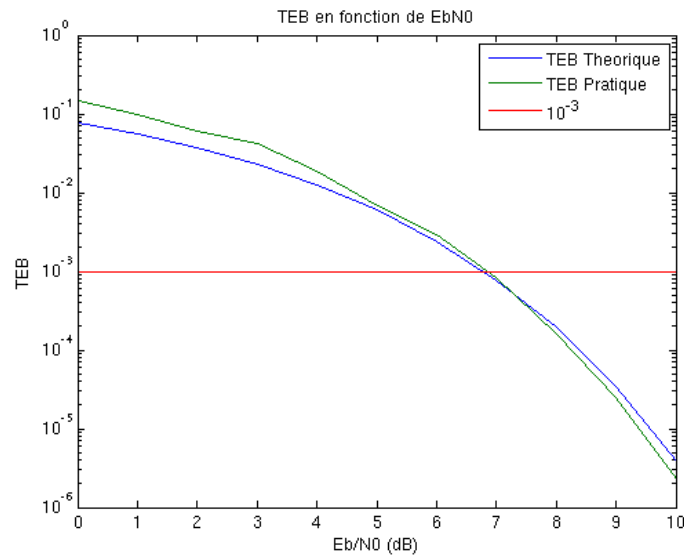


Figure 10: Taux d'Erreur Binaire en fonction du rapport signal sur bruit, avec estimation temporelle et fréquentielle

Traitement / décodage de signaux réels

1 Structure des trames ADS-B

Question 18

- Les trames de position en vol correspondent à des valeurs de FTC de 9 à 18 et 20 à 22.
- Les trames d'identifications correspondent à des valeurs de FTC de 1 à 4.

2 Travail à réaliser

Question 20 Après extraction des informations sur les 21 trames issues d'un même appareil, on peut enregistrer les différentes coordonnées de l'avion, provenant des trames de positions en vol.

Voir pour cela la figure 11

Question 21 D'après des informations issues du site flightradar24.com, l'avion d'adresse 393324 (hexadécimal) appartient à la compagnie Air France (®). C'est un avion de modèle A321 et, d'après sa trajectoire, il se dirigeait vers le nord.

Part III

Temps réel

Cette partie permet de mettre en œuvre les codes créés pour les deux parties précédentes. En fusionnant nos travaux et en utilisant l'antenne fournie, nous sommes capable d'afficher en temps réel des informations à propos des avions autour de l'aéroport de Mérignac.

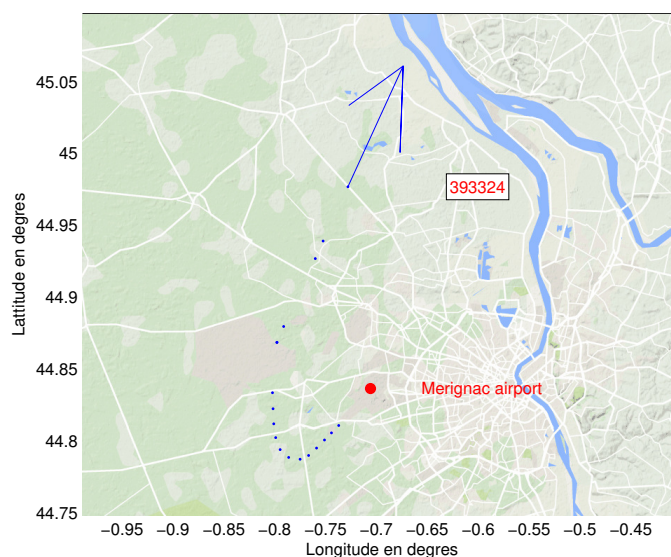


Figure 11: Trajectoire de l'appareil sur une carte Google Maps.

Remarque Nous souhaitons remercier le binôme de Simon et Clément pour les buffers qu'ils nous ont fourni. En effet, cela nous a permis de travailler en temps extra-scolaire sur cette dernière partie.

Remarque 2 À noter que la fusion des codes (parties 1 et 2) n'a pas fonctionné directement. Lorsque nous nous sommes penché sur le cas particulier du temps réel, de nombreuses erreurs sont apparus. Au départ, seules certaines trames étaient affichées (en comparant avec le binôme cité précédemment). Les deux erreurs principales étaient la mauvaise initialisation du filtre p (voir remarque page 9), et une mauvaise implémentation du bloc de décision.

1 Optimisation temps réel

Pour pouvoir traiter les trames reçues en tant réel, nous nous sommes intéressé aux optimisations possibles pour réduire le temps d'exécution du programme.

1.1 Détermination du CRC

Les fonctions proposées par Matlab pour la détection d'erreur CRC sont très simples d'utilisation mais sont quelque peu gourmandes en terme de temps d'exécution (appels à des fonctions tel que `bi2dec` plusieurs centaines de fois).

Nous avons donc cherché à implémenter une fonction, plus rudimentaire, mais moins chronophage. Nous nous sommes inspiré d'une fonction proposée sur MathWorks², qui calcule le CRC pour un polynôme générateur de 16 bits.

En testant avec la fonction d'origine de Matlab, et en activant notre correction d'erreur (voir 2, page 15), on atteint presque 9 secondes passées dans la fonction `Detector`. Avec la nouvelle fonction mise en place exclusivement pour notre cas, le temps passé dans `crc24` est d'une seconde et demie.

1.2 Registre

Pour stocker les données des avions issues des trames, nous avons deux choix possibles.

²<http://in.mathworks.com/matlabcentral/fileexchange/26741-crc-16>

Le premier était de faire un tableau de structure, contenant les informations d'un avion par structure. Le second était de n'avoir qu'une seule structure contenant des vecteurs dont l'indice déterminait l'avion.

Pour faire notre choix nous avons comparé la gestion, par Matlab, de la mémoire selon les deux configurations. Nous les avons donc testées avec exactement les mêmes données : 10 avions avec chaque champ rempli de données similaires, et pour chaque avion, cinq coordonnées.

Name	Size	Bytes	Class
registre_structofvectors	1×1	11808	struct
registre_arrayofstruct	1×10	20832	struct

Il semble que MATLAB enregistre les informations relatives aux structures pour chaque indice du tableau, ce qui prend beaucoup de mémoire. Nous avons donc préféré un unique registre contenant des vecteurs.

1.3 Convolutions

Suite à nos tests avec le "profilier" de Matlab, il s'est avéré que le temps passé dans la fonction `xcorr` était particulièrement important. Nous nous sommes donc attelé à remplacer cette fonction par une convolution (voir Question 17, page 12), puisque :

$$R_{xx}(\tau) = x(\tau) * x^*(-\tau) \quad (46)$$

2 Correction d'une erreur

Lorsque notre code était non fonctionnel, et ne détectait pas toutes les trames possibles, nous avons cherché à corriger un bit d'erreur pour augmenter notre nombre de trames valides découvertes (implémenté au sein du fichier `get_trames.m`).

Pour ce faire, il suffit d'inverser un bit de la trame, et de vérifier à l'aide du CRC si celle ci devient bonne.

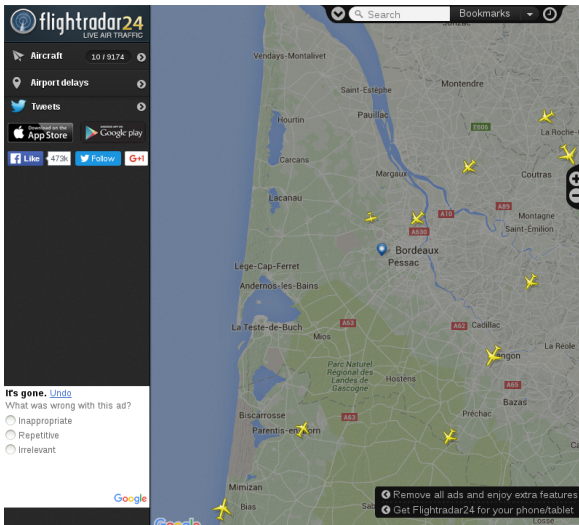
Afin de ne pas surcharger le programme, on va faire en sorte de ne pas tester tous les bits : l'idée est d'attribuer une valeur à chaque bit, correspondant à un poids. Cette valeur est calculée en observant si, au moment d'échantillonnage, un bit est proche de là où la décision a été faite, ou non.

Ce qui nous permet d'appliquer notre algorithme uniquement aux bits avec un poids plus faible (\neq bits de poids fort ou faible du binaire), qui correspond à une probabilité plus élevée d'erreur.

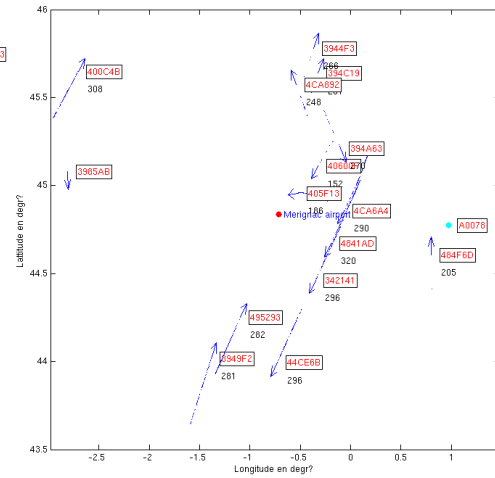
En testant sur les buffers fournis par Simon et Clément, sans correction d'erreur nous obtenons 86 trames valides, et 91 avec. À noter que cette correction augmente le temps d'exécution du programme, et ne peut donc pas être utilisée en temps réel.

3 Vitesse et direction des avions

Nous avons implémenté le décodage des trames de vitesse envoyées par les avions dans le fichier `decode_velocity.m`. Ces trames contiennent des informations sur la vitesse ainsi que sur le cap des avions. Cependant, la valeur du cap fournie par les trames ne donne que l'angle relatif par rapport à sa trajectoire. Pour faciliter le calcul de la direction nous avons préféré utiliser les deux dernières positions de l'avion pour estimer sa direction. Cette opération se fait dans le fichier `display_head.m`. Sur la figure, cela se traduit par une flèche au niveau de la dernière position relevée et de la vitesse affichée à côté de l'adresse OACI de l'avion (voir figure 12b).



(a) Positions des avions d'après flightradar24.com



(b) Positions des avions d'après notre algorithme

Figure 12: Capture des trames ADS-B, à 11h30 (18 déc. 2015) avec notre algorithme et le site flightradar24.com

4 Figures de résultat

En fusionnant nos deux premières parties, et avec nos modifications apportées au code (présentées dans cette partie du rapport), nous pouvons afficher sur une carte, en temps réel, la position des avions. Nos résultats finaux sont présentés en figure 12.

Remarque À noter que, sur la figure 12b (notre capture des positions des avions), la carte de Google n'a pas été affichée. En effet, puisque l'adresse IP publique de l'école est la même pour tous les étudiants, elle a atteint le quota maximal de téléchargement de la carte via le service de Google.